

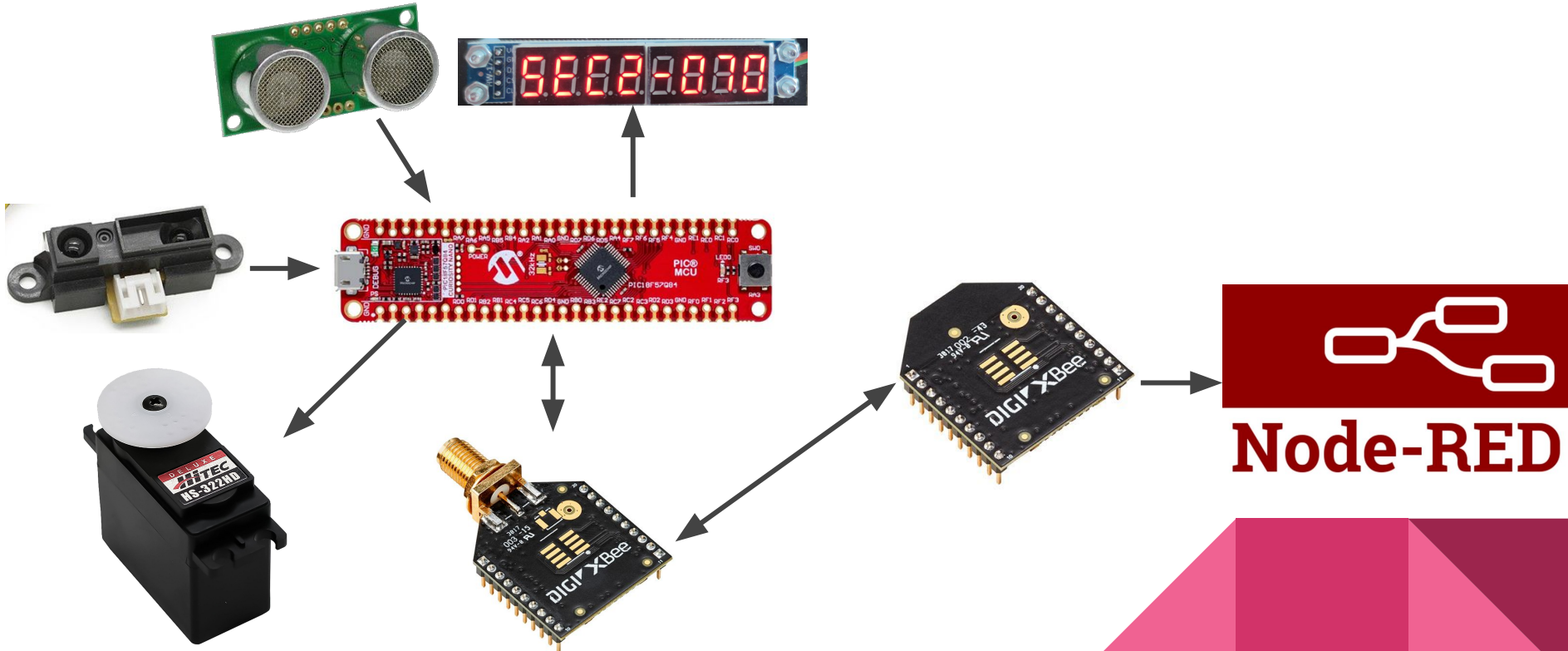
# SAE Contrôle d'une tête rotative de détection d'obstacles pour robot mobile.

DEVILLERS Tom | PATERNOTTE Mattéo

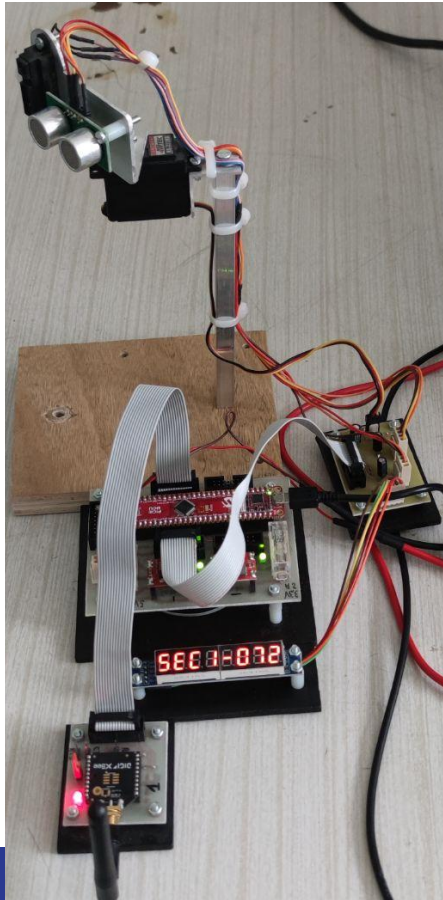
# Plan de présentation :

- Présentation du système
- Présentation du fonctionnement
- Télémètre Infrarouge
- Sonar SRF05
- Visualisation d'un afficheur
- Servo moteur
- Communication Radio
- Réalisation d'une interface graphique
- Conclusion

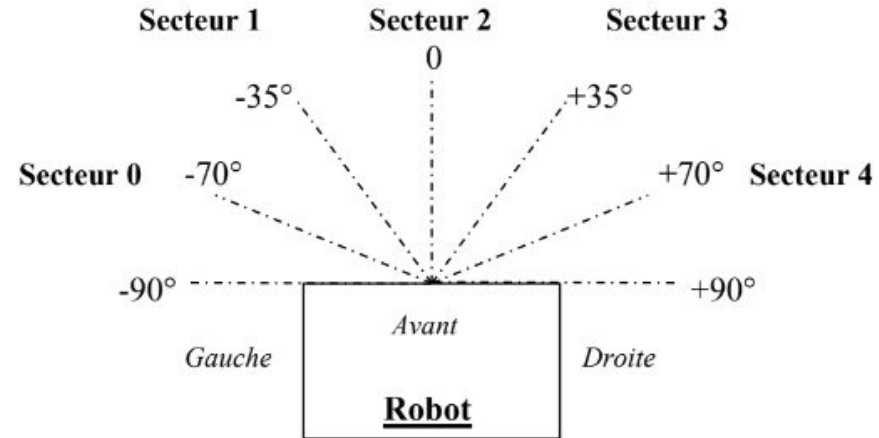
# Présentation du système



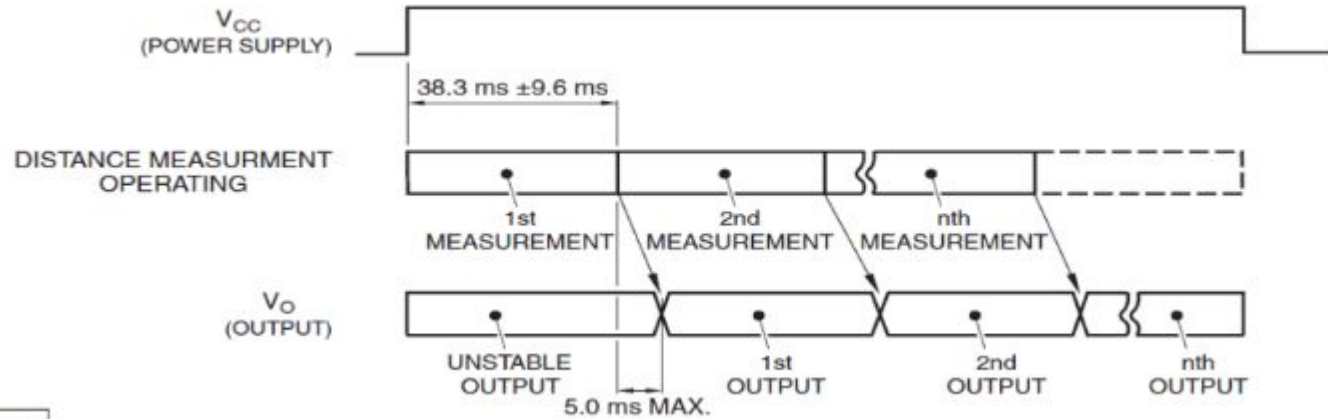
# Présentation du fonctionnement



```
if(secteur==0&&sens_rotations==1&& reset==0){  
    for(angle=0;angle<=8;angle++){  
        balayage=70-angle*3.05;  
        set_pwm3_duty(balayage);  
        delay_ms(200);  
        mesure(angle);  
        reset=1;  
        xbee_a(affichage);  
    }  
}
```

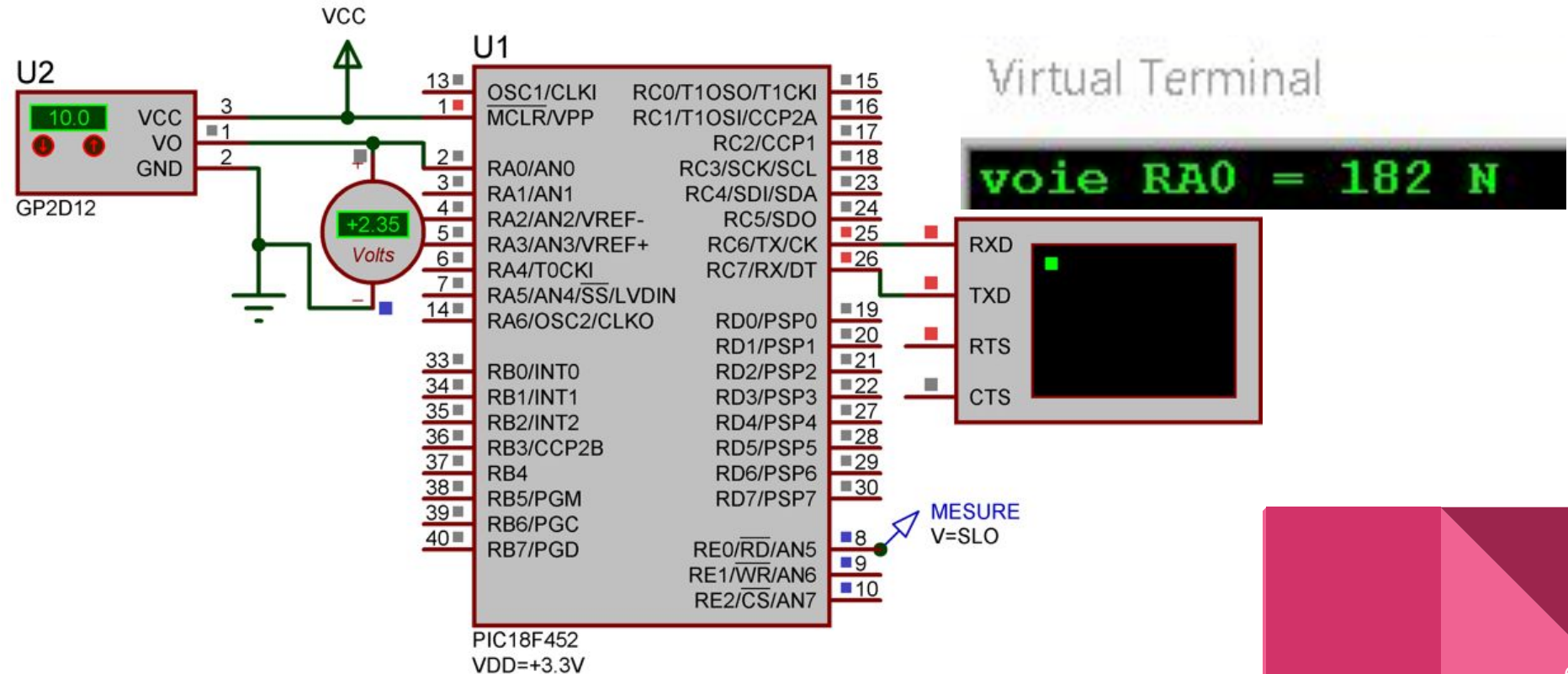


# Télémètres infrarouge (GP2D12)



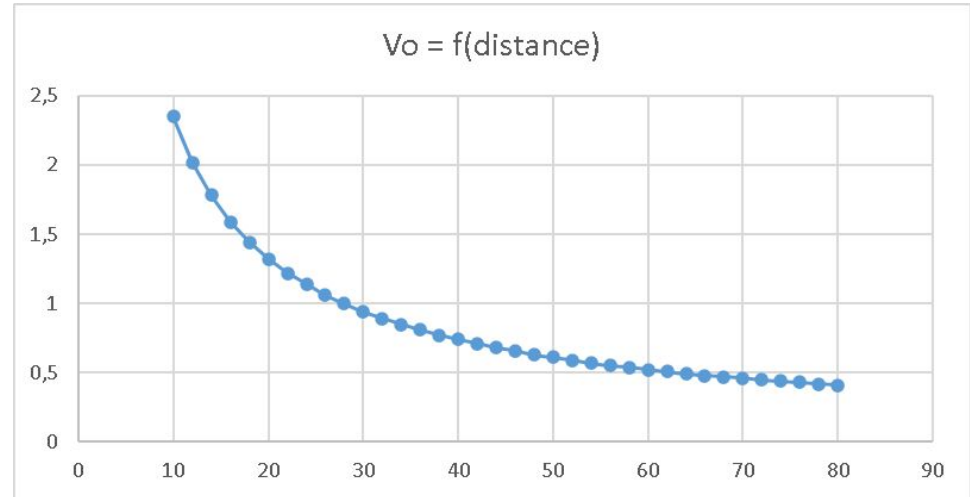
PIN	SIGNAL NAME
①	$V_O$
②	GND
③	$V_{CC}$

# Télémètres infrarouge (GP2D12)

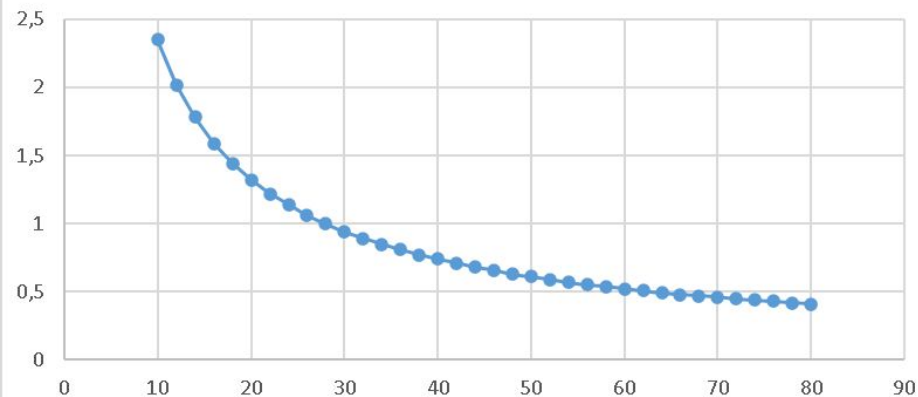


# Télémètres infrarouge (GP2D12)

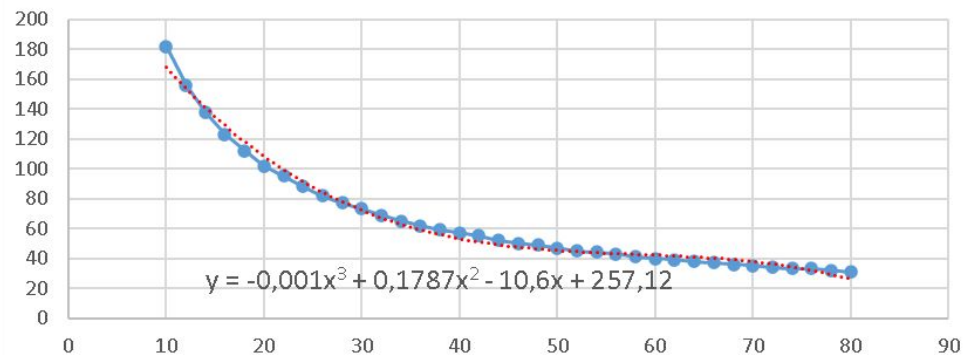
Mesure avec proteus		8bits / Vref=3,3V
DISTANCE (cm)	Volts	Nombre binaire
10	2,35	182
12	2,02	156
14	1,78	138
16	1,59	123
18	1,44	112
20	1,32	102
22	1,22	95
24	1,14	88
26	1,06	82
28	1	77
30	0,94	73
32	0,89	69
34	0,85	65
36	0,81	62
38	0,77	59
40	0,74	57
42	0,71	55
44	0,68	52



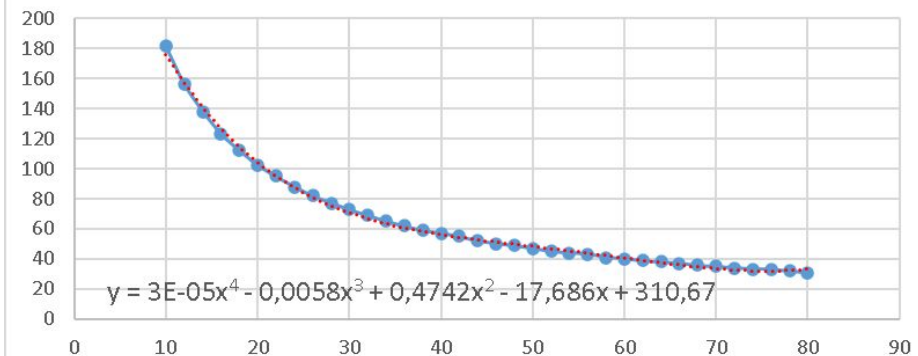
$V_0 = f(\text{distance})$



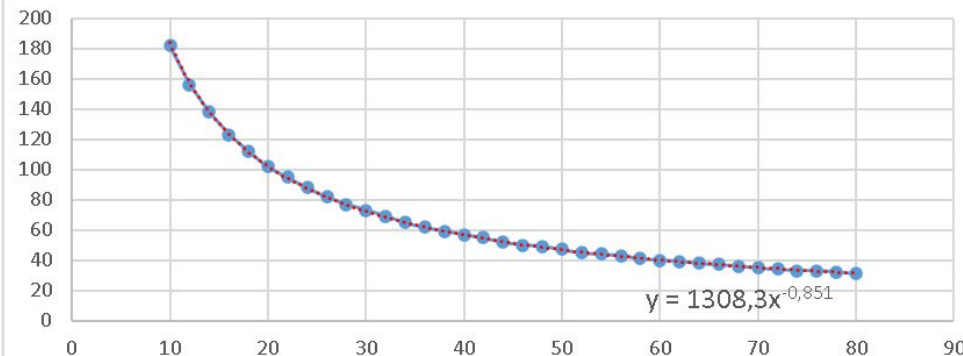
Distance en cm =  $f(\text{Nadc})$   
régression polynomial d'ordre 3



Distance en cm =  $f(\text{Nadc})$   
régression polynomial d'ordre 4



Distance en cm =  $f(\text{Nadc})$   
régression de puissance





# Télémètres infrarouge (GP2D12)

Mesure avec proteus		8bits / Vref=3,3V	Distance en cm = f(Nadc)		
DISTANCE (cm)	Volts	Nombre binaire	Polynome ordre 3	Polynome ordre 4	Puissance
10	2,35	182	-1781,3892	10749,50568	15,60941367
12	2,02	156	-844,0528	4839,79928	17,79747158
14	1,78	138	-430,5892	2538,06728	19,75469091
16	1,59	123	-203,9947	1383,03443	21,78703302
18	1,44	112	-93,3952	850,17848	23,59515334
20	1,32	102	-26,0932	532,56488	25,54986414
22	1,22	95	5,5125	380,89875	27,14341916
24	1,14	88	26,7008	273,05528	28,97027297
26	1,06	82	38,1308	207,36968	30,76463402
28	1	77	43,8993	167,07963	32,45665314
30	0,94	73	46,5953	142,25243	33,96405906
32	0,89	69	48,0017	122,66363	35,63253867
34	0,85	65	48,5025	107,26875	37,49022742
36	0,81	62	48,5148	97,95048	39,02851379
38	0,77	59	48,3957	90,20883	40,71104774
40	0,74	57	48,3233	85,80443	41,92352959
42	0,71	55	48,3125	81,93875	43,21740566
44	0,68	52	48,5168	77,05688	45,33028984

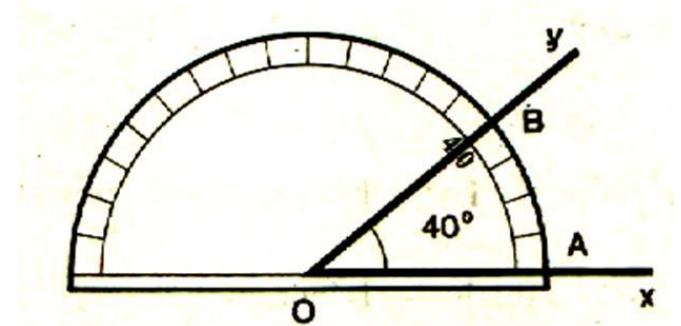
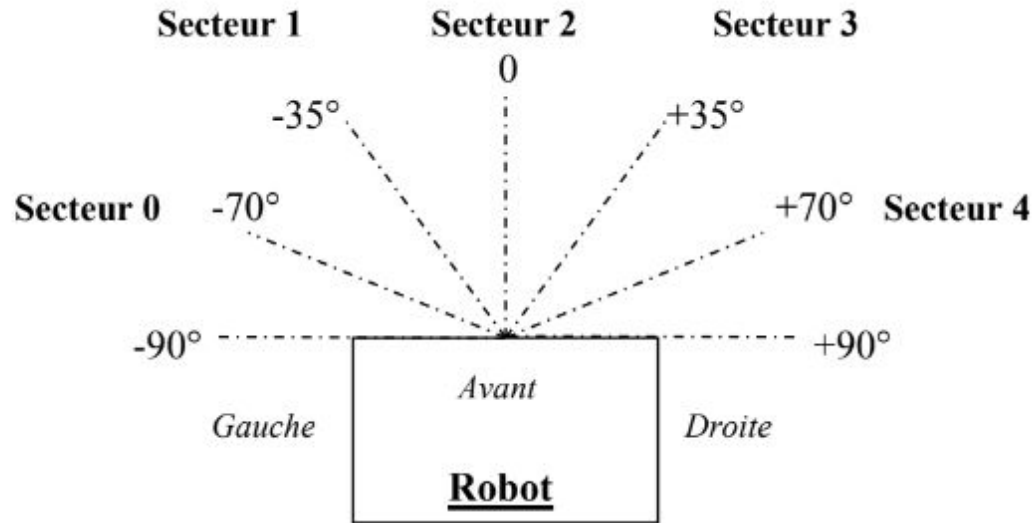
Calcul	Temps
Polynome ordre 3	726us
Polynome ordre 4	1240us
Puissance	4400us

$$D(\text{cm}) = 1308,3 * \text{Nadc}^{-0,851}$$

## Télémètres infrarouge (GP2D12)

```
void mesure(int angle)
{
    nadc = read_adc(); // demande de conversion, att
    delay_ms(53); // Attendre le temps maximum pour f
    calcul2=1308.3*pow(nadc, -0.851); // Calcul de la
    if(calcul2>10&&calcul2<70){ // Si la distance mes
    fprintf(DEBUG,"n°%u ; %lf \n\r",angle,calcul2);/
    }
}
```

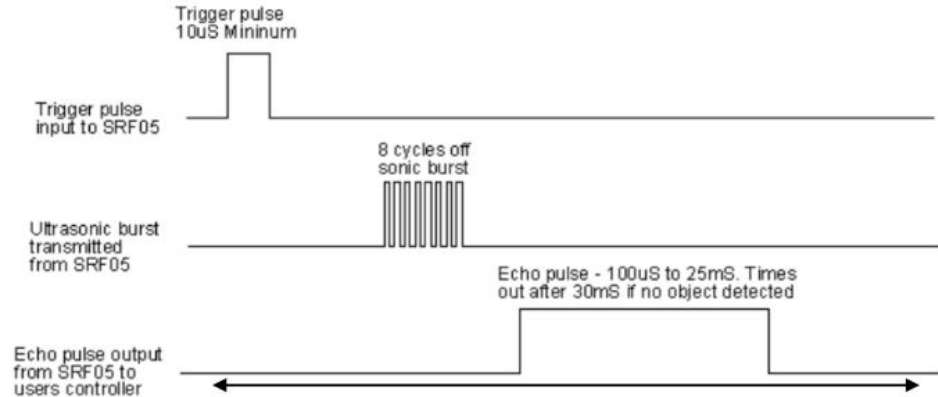
# Sonar SRF05



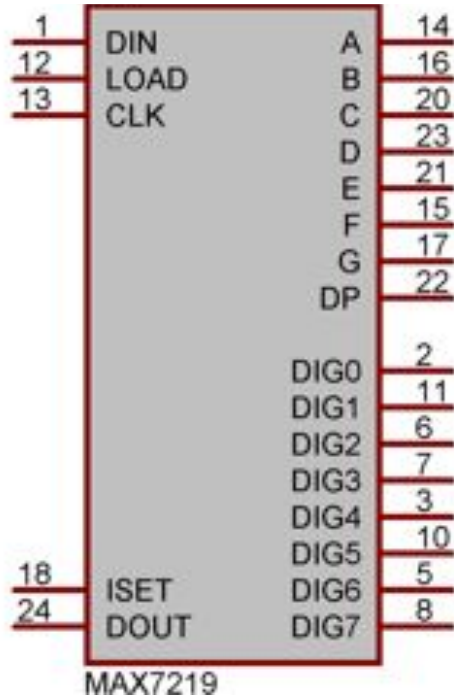
# Programmation :

```
void debut()  
{  
  output_high(Start);  
  delay_us(10);  
  output_low(Start);  
  set_timer1(0);  
}
```

```
void fonctionnement()  
{  
  debut();  
  while(input(EPO)==0 && flag_stop==0){  
  }  
  while(input(EPO)==1 && flag_stop==0){  
  }  
  if(flag_stop==0)  
  {  
    sonar=get_timer1();  
    distance = (float)sonar/2*340/20000;  
    fprintf(DEBUG,"distance SONAR: %f \r\n",distance);  
  }  
  else  
  {  
    fprintf(DEBUG," aucun obstacle \n\r");  
    flag_stop=0;  
  }  
}
```



# Visualisation d'un afficheur :



Digits : 8 7 6 5 4 3 2 1



- Programmation → Librairie du MAX7219
- Deux modes : décodage BCD  
une gestion binaire des segments.

```
void affichage_7219(float distance,int secteur)
```

```
{  
  centaine=(int)distance/100;  
  dizaine((((distance+100)/10));  
  dizaine=dizaine%10;  
  unite=(int)distance%10;
```

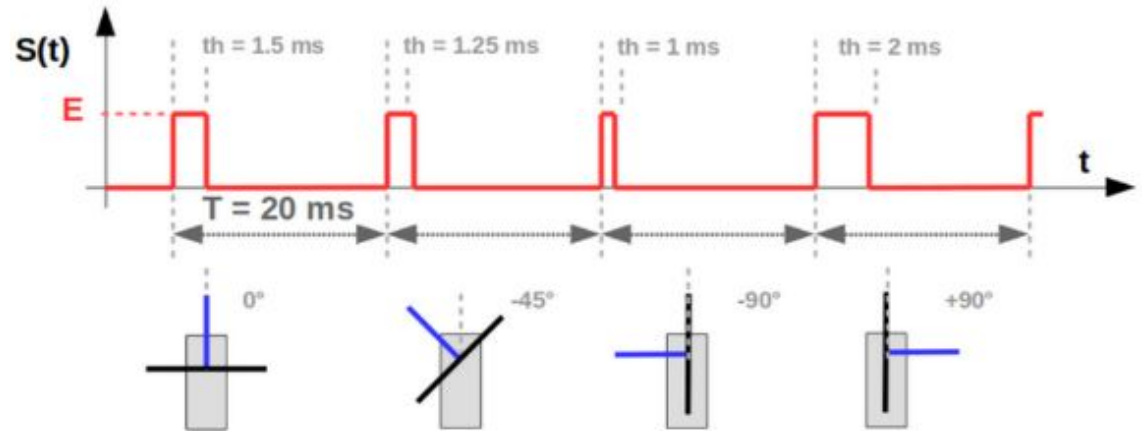


```
decode7219(0b10010111);// 1 MODE BCD // 0 MODE GESTION DES SEGMENT DE FAÇONS BINAIRE
```

```
write7219(8, 5);// on affiche 5 positions 8  
write7219(7, 0b1001111);// on affiche E positions 7  
write7219(6, 0b1001110);// on affiche C positions 6  
write7219(5, secteur);// on affiche le chiffre du secteur position 5  
write7219(4, 0b0000001);// on affiche le tirret du secteur position 4  
write7219(3, centaine );// on affiche la dizaine de la distance position 3  
write7219(2, dizaine);// on affiche unite de la distance position 2  
write7219(1, unite);// on affiche unite de la distance position 2  
}
```



# Servo moteur :



$$t1 = -90^\circ$$

$$t2 = 90^\circ$$

$$t1 = \frac{0.001}{\frac{1}{8 \times 10^6} \times 128} = 62.5$$

$$t2 = \frac{0.002}{\frac{1}{8 \times 10^6} \times 128} = 125$$

# Programmation :

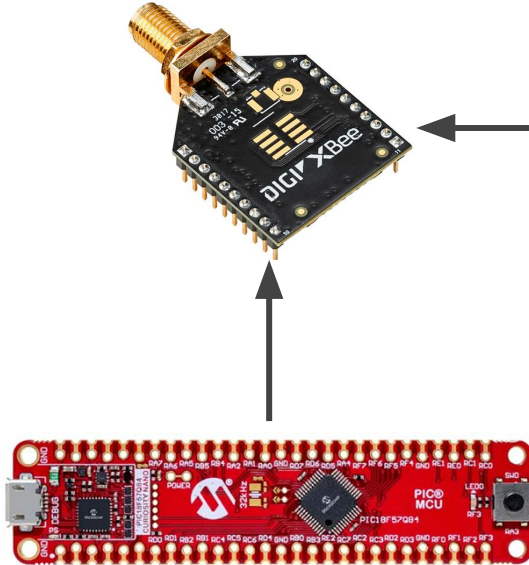
```
if(flag_objet_detecte==0)
{
    switch(secteur){
        case 4:
            sens_rotations=1;
            set_pwm3_duty((int16)143);//+70 degré
            break;
        case 3:
            set_pwm3_duty((int16)121);//+35 degré
            break;
        case 2:
            set_pwm3_duty((int16)100);// 0 degré
            break;
        case 1:
            set_pwm3_duty((int16)79);//-35 degré
            break;
        case 0:
            sens_rotations=0;
            set_pwm3_duty((int16)58);//-70 degré
            break;
    }
}
```

degré	nombre
-90	45
-70	58
-55	66
-50	70
-35	79
-15	91
0	100
15	109
35	121
50	130
55	133
70	143
90	155

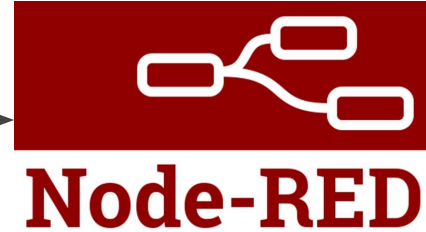


# Communication Radio :

**Coordinateur :**



**Routeur :**





# Réalisation d'une interface graphique :

Noeuds

Installer


Rechercher le noeud


 node-red

 3.1.7

> 50 noeuds


En cours d'utilisation


 node-red-dashboard

 3.6.5

> 21 noeuds

En cours d'utilisation

 node-red-node-serialport

 2.0.2

> 5 noeuds

En cours d'utilisation

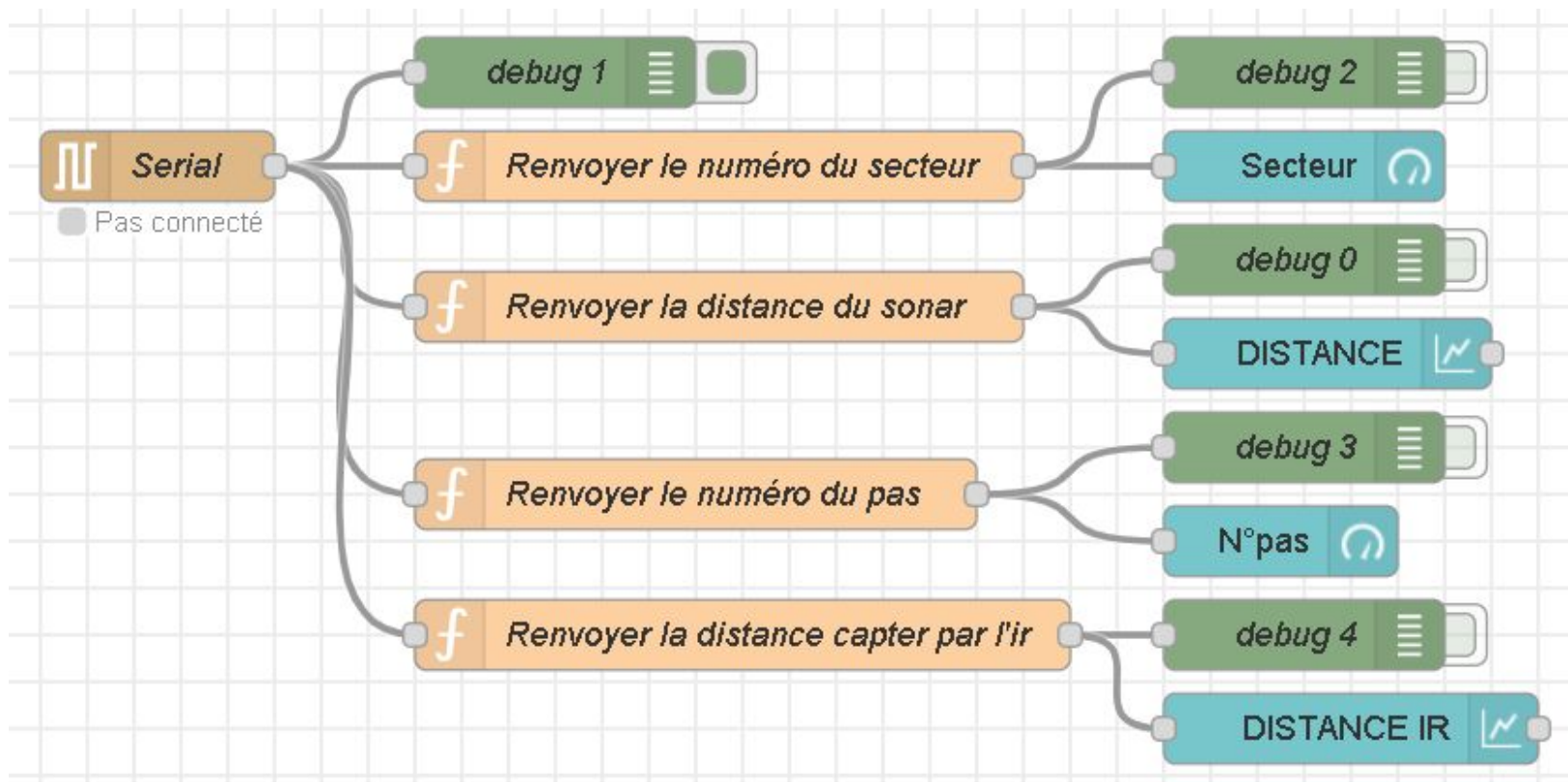
```
02/04/2024 10:34:35  noeud: debug 1  
msg.payload : string[19]
```

```
▶ "  
secteur n 1;59.13↵"
```

```
02/04/2024 10:34:35  noeud: debug 1  
msg.payload : string[12]
```

```
▶ "  
pas n 0;00↵"
```

# Réalisation d'une interface graphique :





# Conclusion

DEVILLERS Tom | PATERNOTTE Mattéo